Before asking the question, how do I build AGI, you first must ask the question, what would I recognize as AGI?

I suspect there's not a "real" answer for this. Humans can be fooled both ways, into thinking simple things have "intelligence" and "life", and into thinking complex things don't. Though they usually aren't fooled for too long, at least with the simple things.

I define AGI as *a machine capable of taking (almost) any human job*. This says a few things about what it must be like.

Mostly it implies that this thing must be human like enough that others humans quickly accept it as ~human. Just like L2 driving systems will incrementally win self driving cars, there won't be a huge overnight switch to the robo(taxi) economy. AGI, to be recognized as such, will have to interoperate with humans, just as self driving cars will have to interoperate with human driven ones.

This implies the following:

- It must have eyes and ears. While some humans don't, they can't do every job.
- It must fit in a similar space to a human. Like it or not, it's what the world is designed for.
- It must have a similar (or better) charge cycle, 16 hours on for 8 hours off.

But those are the obvious ones:

- It must (somewhat) remember the past.
- It must have goal directed behavior.
- It must be able to communicate efficiently with humans.

Most importantly though, it must **be able to learn**. All humans in all jobs need this skill.

This is Kenny, a comma body. He appeared on my LinkedIn asking for a controls engineer to help him stand.

Though it occurs to me that it would be pointless. Nobody doubts we could make a model and write MPC to control it. We could probably also build a simulation environment, do RL in the simulation, then transfer it over. But what happens when we encounter a bump? Gravel? Snow? A human trying to knock it over? A banana peel? Did you put all those things in your simulator?

The only way forward is on device learning.

Here’s a paper from Berkeley, Learning to Walk in 20 Minutes where they got a robot dog to walk training live (mostly, laptop offboard) on the device.

I think there can be good universal code that can run on any robot and manage to learn to control it without breaking it. Unlike simulation, “resetting” can be much more expensive in the real world, so you’ll need to take this into account. There are bad states, and there are BAD states.

For a long time on my whiteboard I had the phrase “What is the human reward function?” I think this is a question we’ll have to answer, the basis for it is in our DNA, I don’t think there’s a way to learn it aside from millennia of multi agent survival competition.

To start, I think simple reward functions would work, like standing or movement. I imagine you’ll end up adding things like curiosity and self preservation. It might not end up being too hard.

If you are interested in this problem, come work at comma.ai. I imagine a day when you can drop openpilot on any robot and it can learn to control it.

Update after HN discussion (wow you found this fast, I wasn’t finished writing!)

There’s definitely a “copy” component of the human reward function, though this comment explains why it’s incomplete. This copy ability is another reason that embodiment helps, the machine can develop “mirror neurons” and be able to learn watching people do things on YouTube like the rest of us.

Sim embodiment can be fine, but humanoid sim embodiment, and your sim better be super super good. Perhaps Carmack thinks coding this simulator is easier than I do, I think robotics is easier. And unless you want to be stuck in a simulated world, you have to do robotics anyway.

I suspect artificial life learning looks like this (since this is what human learning looks like):

- Learn the basics on device IRL with some hand coded reward functions

- Learn about, say, cooking and cleaning by watching people
- Perfect the skills learning on device IRL by doing

At comma we discuss three paradigms of models that matches the above:

- one was hand coded lanes and cars, OG openpilot up to ~0.6
- two is where we are currently, with “supervised” learning in simulation, full alpha e2e in 0.9
- three is reinforcement learning on the world (must have decent full e2e policy first)

You can’t go straight from one to three, because you cannot backprop through a human intervention event to a bad location of a lane or car. But if your model is outputting a path, you can backprop the human intervention to that.

There’s not even a way to specify the complete reward function for driving, never mind for life. You have to do RL with other real agents at some point.

And to respond to the person *scared* of asking about the human reward function, I’m asking it in a much more abstract way than I think you imagine I am.

Obviously the human reward function is learned, imagine showing a fiending addict a line and watching their reward centers light up (did you like the picture?). You don't even have to give them the drugs. Lines of white powder are obviously not specified in our DNA. (and if you think it's because the drugs are directly acting, do you not think this also works with money?)

But there is an underlying basis for the (learned) human reward function, the meta reward function say? This is what's in our DNA, and this is what I want to capture. And I'm almost sure it's multi part, different humans seem to have different weights on the parts. Parts may include:

- autoregression
- curiosity
- behavioral cloning
- doing what others want them to (is this emergent?)

Perhaps the first two are combined to maximize the derivative of compression.